



本ドキュメントはCypress (サイプレス) 製品に関する情報が記載されております。本ドキュメントには、仕様の開発元企業として「スパンション」, 「Spansion」, 「富士通」または「Fujitsu」の名が記載されておりますが、これらの製品は Cypress が新規および既存のお客様に引き続き提供してまいります。

商品仕様の継続性について

Cypress 製品として提供することに伴う商品仕様としての変更はなく、ドキュメントとしての変更もありません。また本ページのお知らせは、変更情報として追記いたしません。本ドキュメントに変更情報が記載されている場合、それは本お知らせを除いた前版からの変更点です。なお、今後改訂は必要に応じて行われますが、その際の変更内容は改訂後のドキュメントに記載いたします。

オーダ型格および品名について

Cypress は既存のオーダ型格および品名を引き続きサポートいたします。これらの製品をご注文の際は、このドキュメントに記載されているオーダ型格および品名をご使用ください。

詳しいお問い合わせ先

Cypress 製品およびそのソリューションの詳細につきましては、お近くの営業所へお問い合わせください。

サイプレスについて

サイプレス (銘柄コード: CY) は、車載や産業機器、ネットワーキング プラットフォームから高機能民生機器およびモバイル機器まで、今日の最先端組み込みシステム向けに高性能で高品質のソリューションを提供します。NOR フラッシュ メモリや F-RAMTM、SRAM、TraveoTM マイクロコントローラー、業界唯一の PSoC[®] プログラマブル システムオンチップ ソリューション、アナログおよび PMIC Power Management IC、CapSense[®] 静電容量タッチセンシング コントローラー、Wireless BLE Bluetooth[®] Low-Energy、USB コネクティビティ ソリューションなど、幅広い差別化製品ポートフォリオを、一貫した革新性と業界最高クラスの技術サポート、比類のないシステム バリューとともにグローバルに提供します。

FR Family
32-BIT MICROCONTROLLER
MB91260B/MB91265A

積和演算回路の使用方法

注意事項

- 本資料の記載内容は、予告なしに変更することがありますので、ご用命の際は営業部門にご確認ください。
- 本資料に記載された動作概要や応用回路例は、半導体デバイスの標準的な動作や使い方を示したもので、実際に使用する機器での動作を保証するものではありません。従いまして、これらを使用するにあたってはお客様の責任において機器の設計を行ってください。これらの使用に起因する損害などについては、当社はその責任を負いません。
- 本資料に記載された動作概要・回路図を含む技術情報は、当社もしくは第三者の特許権、著作権等の知的財産権やその他の権利の使用権または実施権の許諾を意味するものではありません。また、これらの使用について、第三者の知的財産権やその他の権利の実施ができることの保証を行うものではありません。したがって、これらの使用に起因する第三者の知的財産権やその他の権利の侵害について、当社はその責任を負いません。
- 本資料に記載された製品は、通常の産業用、一般事務用、パーソナル用、家庭用などの一般的用途に使用されることを意図して設計・製造されています。極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、社会的に重大な影響を与えかつ直接生命・身体に対する重大な危険性を伴う用途（原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療機器、兵器システムにおけるミサイル発射制御をいう）、ならびに極めて高い信頼性が要求される用途（海底中継器、宇宙衛星をいう）に使用されるよう設計・製造されたものではありません。したがって、これらの用途にご使用をお考えのお客様は、必ず事前に営業部門までご相談ください。ご相談なく使用されたことにより発生した損害などについては、責任を負いかねますのでご了承ください。
- 半導体デバイスはある確率で故障が発生します。当社半導体デバイスが故障しても、結果的に人身事故、火災事故、社会的な損害を生じさせないように、お客様は、装置の冗長設計、延焼対策設計、過電流防止対策設計、誤動作防止設計などの安全設計をお願いします。
- 本資料に記載された製品を輸出または提供する場合は、外国為替及び外国貿易法および米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとりください。
- 本書に記載されている社名および製品名などの固有名詞は、各社の商標または登録商標です。

Copyright© 2008 FUJITSU MICROELECTRONICS LIMITED all rights reserved

改版履歴

版数	日付	内容
1.0 版	2005.12.23	新規作成
1.1 版	2008.01.21	新フォーマットに修正
1.2 版	2008.02.25	P7 の誤記を修正(図 1.2.3→図 2.2.3)

目次

注意事項	1
改版履歴	2
目次	3
1 はじめに.....	4
2 DSPマクロ ASMコンバータを使用するための準備.....	5
2.1 インストール方法.....	5
2.2 Softune Workbenchへの登録	5
3 DSPマクロを使用する際のプログラムの作成手順	11
付録	15
A メモリの配置	15
B MB91260Bシリーズ DSPメモリ空間の例	16

1 はじめに

この手引きは、富士通の32ビット MCU MB91260B および MB91265A シリーズに搭載されている、16ビット積和演算マクロ(DSP マクロ)を使用する際の ASM コンバータの導入、利用方法についてまとめました。以下断りのない限り、MB91260B シリーズを例として説明します。

DSP マクロ ASM コンバータの仕様については、別途「DSP マクロ ASM コンバータマニュアル」をご参照下さい。

2 DSP マクロ ASM コンバータを使用するための準備

2.1 インストール方法

DSP マクロ ASM コンバータのインストール方法を以下に示します。

あらかじめ、Softune Workbench はインストールしておいてください。

フォルダ内の「SETUP.exe」ファイルを実行すると、インストーラが起動します。

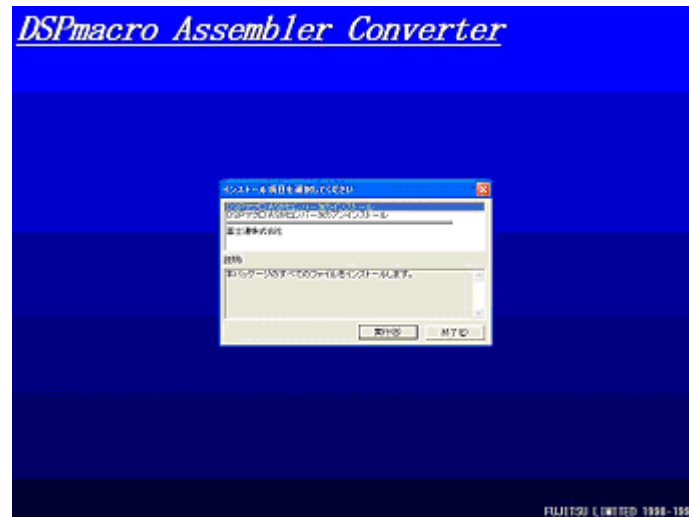


図 2.1 インストーラ起動画面

「DSP マクロ ASM コンバータのインストール」を選択し、インストールフォルダを指定することでインストールが完了します。

(Softune インストールディレクトリ内へのインストールを推奨いたします。)

2.2 Softune Workbench への登録

DSPマクロASMコンバータをSoftune Workbenchに登録する手順、登録内容について説明します。

1) 登録手順

まず最初に、Softune Workbench を起動します。

次にメニューの[環境(S)]から[ツールの設定(O)…]を選択します。(図2.2.1)

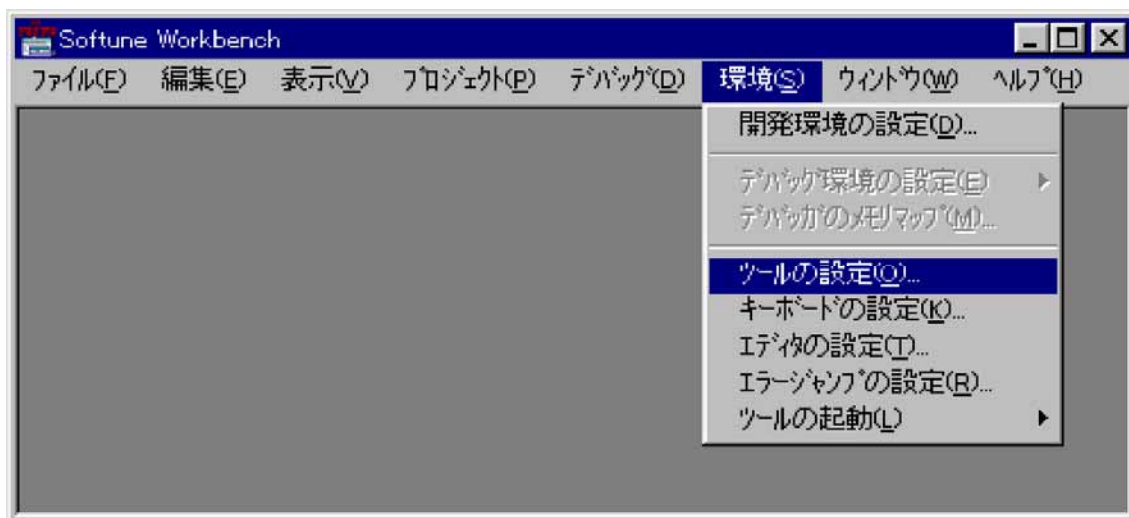


図2.2.1 ツールの設定

上記の操作を行うと、「図2.2.2 ツールの設定ダイアログ」で示すウインドウが表示されますので、ここで必要な情報を入力後、[設定(E)]を選択する事により、DSPマクロASMコンバータを登録することができます。



図2.2.2 ツールの設定ダイアログ

2) 登録内容

「図2.2.2 ツールの設定ダイアログ」で登録する内容について説明します。

【タイトル】

DSPマクロASMコンバータのタイトル名を指定します。

ツールの起動時は、ここで指定したタイトル名を選択します。

【実行ファイル名】

DSPマクロASMコンバータの実行形式ファイル名を指定します。

DSPマクロASMコンバータのインストールディレクトリを環境変数PATHに設定していない場合は、フルパス名で指定して下さい。

【オプション】

DSPマクロASMコンバータ実行時の入力ソースファイル名、および、起動時オプションを指定します。オプション詳細については「DSPマクロ ASMコンバータマニュアル」の起動時オプションを参照してください。

【実行時のディレクトリ】

DSPマクロASMコンバータ実行時のディレクトリを指定します。

【実行時に追加オプションを指定】

チェックありに設定すると、ツールの起動時に追加パラメータの設定ダイアログ(図2.2.3)が表示され、追加オプションを指定する事ができます。

前述の【オプション】で、必要なオプションを設定した場合は、チェックなしに設定して下さい。

【アウトプットウィンドウを使用】

チェックありに設定すると、Softune Workbenchのアウトプットウィンドウにツールの実行結果を出力することができます。Softune Workbenchのエラージャンプ機能を利用する場合はチェックありに設定してください。

【ツールリスト】

登録時は必要ありません。

登録済みツールの登録内容の修正、または登録を削除する場合に使用します。

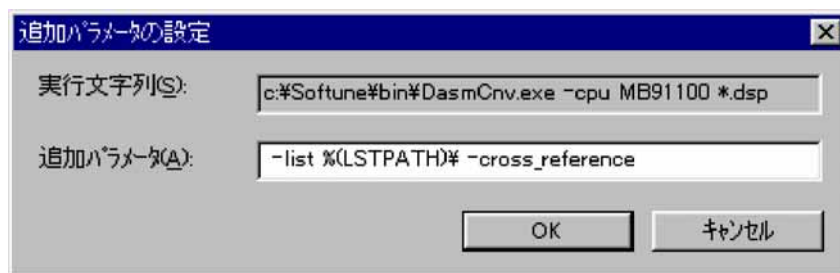


図2.2.3 追加パラメータの設定ダイアログ

注意:

追加パラメータを指定する場合は、先頭に1つ以上の空白を入れて下さい。

例) 追加パラメータ(A):

(_は空白)

3) 登録例

DSPマクロASMコンバータの登録例を記します。

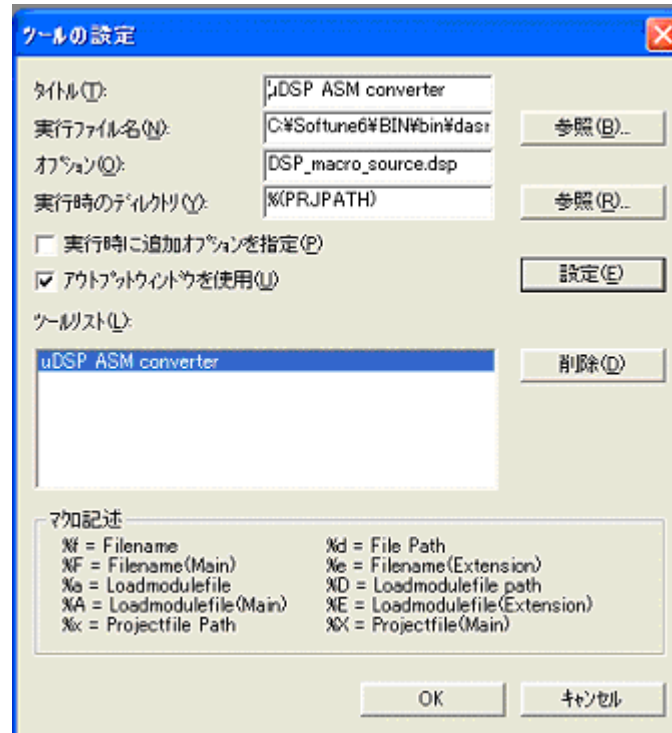


図2.3.1 登録例

- | | | |
|------------------|---|-----------------------------------|
| 【タイトル】 | “uDSP ASM converter” | |
| 【実行ファイル名】 | “C:¥Softune6¥BIN¥bin¥dasm911.exe” | (DSPマクロインストールディレクトリ¥ dasm911.exe) |
| 【オプション】 | “DSP_macro_source.dsp”
(DSPマクロ用アセンブラ記述ファイル名) | |
| 【実行時のディレクトリ】 | % (PRJPATH) | |
| 【実行時に追加オプションを指定】 | チェックなし | |
| 【アウトプットウィンドウを使用】 | チェックあり | |

参考:

オプションに設定するソースファイル名は、ワイルドカード(*.dsp)で指定します。
また、実行時のディレクトリには、プロジェクトのディレクトリ(% (PRJPATH))を指定しています。このように設定することにより、プロジェクトディレクトリ下の全てのDSPソースプログラム(*.dsp)に対し、DSPマクロASMコンバータを実行することができます。

4) ツールの起動

ツールの設定ダイアログで登録したツールを起動するには、メニューの[環境(S)]から[ツールの起動(L)]を表示し、登録したツールのタイトル名を選択します。(図2.4.1)

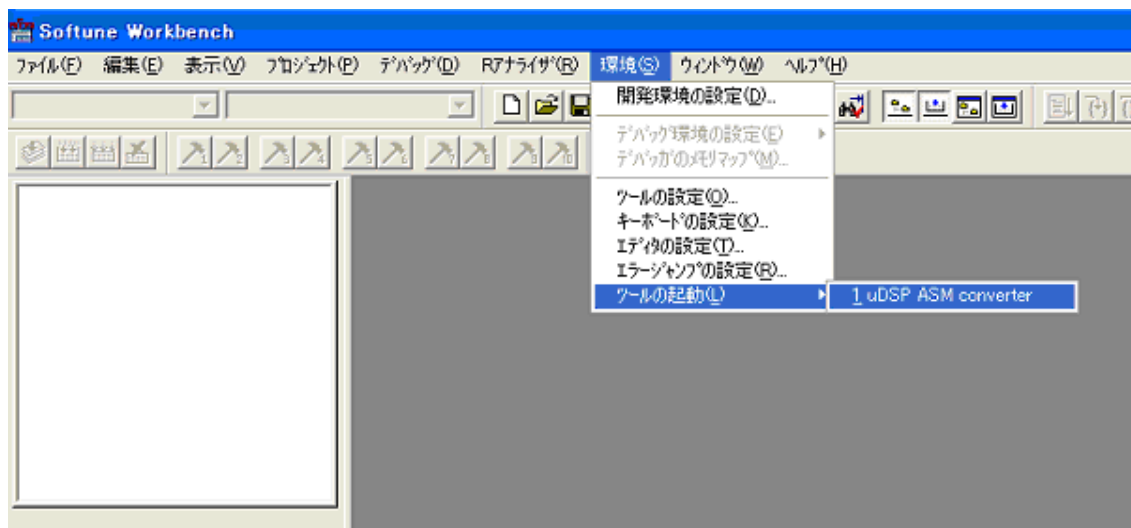


図2.4.1 ツールの起動

3 DSP マクロを使用する際のプログラムの作成手順

DSP マクロ ASM コンバータを使用し、Softune 形式のオブジェクトを作成するまでの流れを図 3.1 に示します。以後、DSP マクロ ASM コンバータを「ASM コンバータ」と記述します。

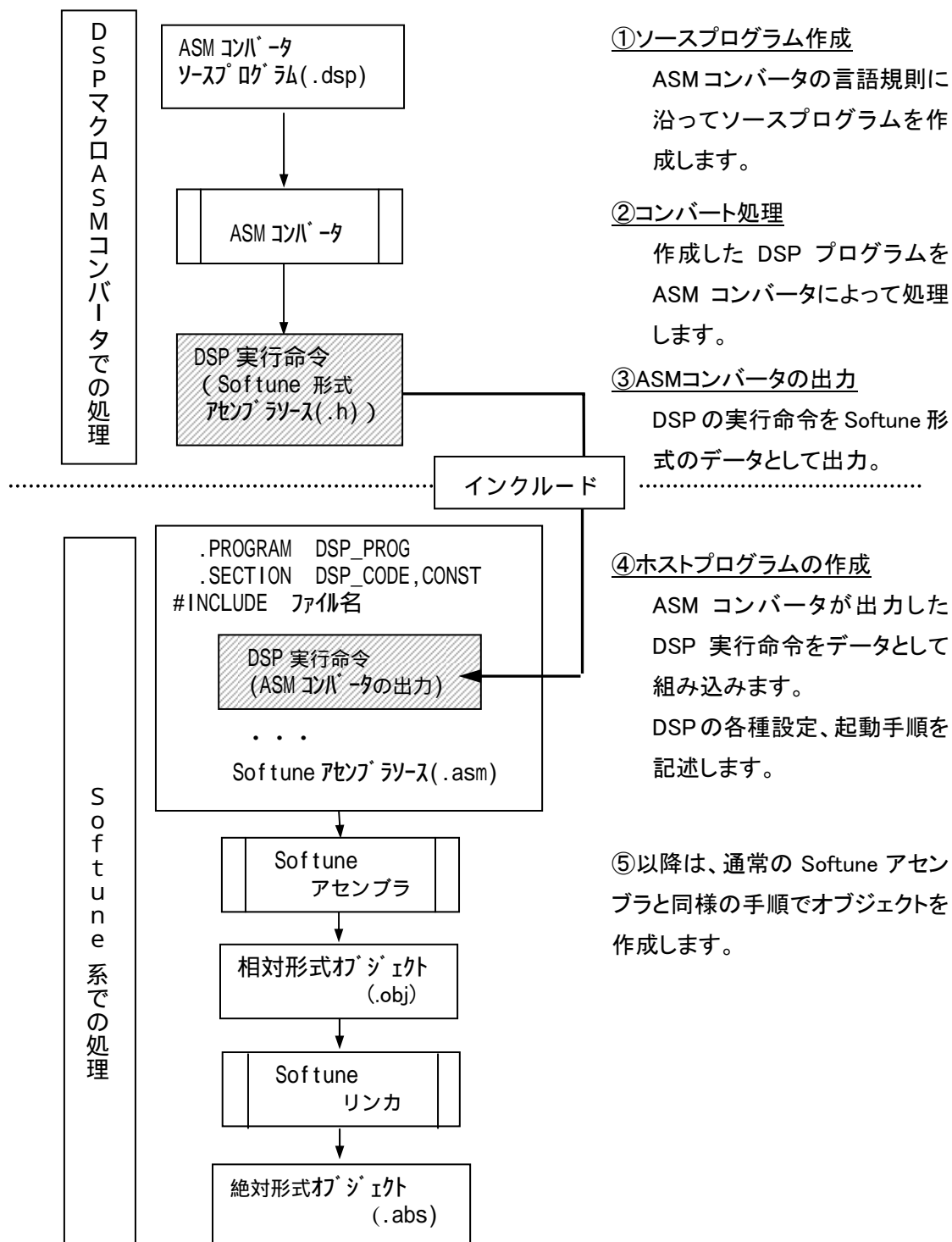


図 3.1 DSP プログラムの作成の流れ

①プロジェクトファイルの準備、およびソースプログラムの作成

プロジェクトファイルは、通常のSoftune Workbenchでの手順と同様になります。

DSPマクロ用のファイル拡張子名は *.dsp としてください。

*.dspファイルは、プロジェクトのメンバ登録してもSoftuneのメイク/ビルドには影響を与えません。

エディット作業の効率化のため、プロジェクトの登録をお勧めします。

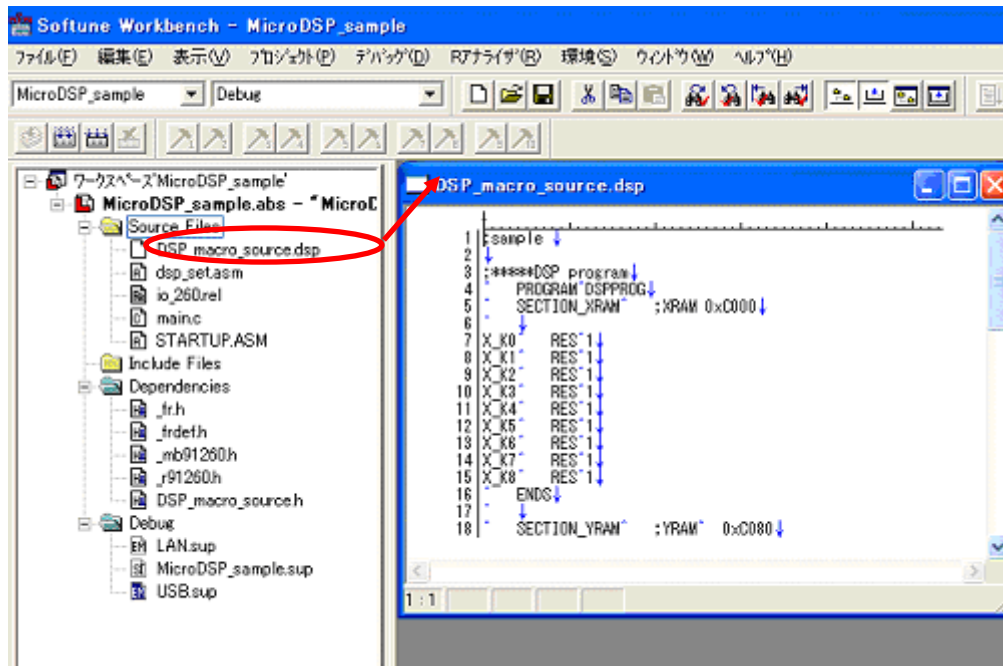


図3.2 *.dspファイルのプロジェクトファイルの登録例

*.dspファイルは、「DSPマクロ ASMコンバータマニュアル」の言語規則に沿ってDSP用ソースプログラムを作成してください。（サンプルプログラム DSP_macro_source.dspを参照下さい。）

注意:

プロジェクトのメイク/ビルド時、DSPソースプログラムとDSPマクロASMコンバータにより生成されたインクルードファイルの依存関係は考慮されません。

DSPソースプログラムを修正した場合は、メイク/ビルドを行う前に、必ず、ツールの設定で登録したDSPマクロASMコンバータを起動し、インクルードファイルを最新のものにしておく必要があります。

②コンバート処理

.dsp ファイルの編集が終了したら、ASM コンバータを起動して Softune 形式アセンブラソース(*.h)を作成します。

起動方法は、2章 4)ツールの起動 を参照してください。

③ASMコンバータの出力

②を実行することで、Softune形式アセンブラソース(*.h)が作成されます。

図 3.3 に *.dsp ファイルの変換結果である DSP_macro_source.h*.h ファイルを示します。

```

/*****
/* DSP Macro I-RAM Section Data
/*
/*   ASM File : DSP_macro_source.dsp
/*
/*
/*           Code Data           ;locate: code   ope1, ope2
/*****
DSP_START:
.DATA.H      1100000000000011B   /* 00H :
.DATA.H      1000000001000100B   /* 01H : MAC   00H,  03H
.DATA.H      10000000010000101B  /* 02H : MAC   01H,  04H
.DATA.H      0111001001000000B   /* 03H : STYIH 02H,  05H
                                /* 03H : STYIH 00H,  Q8

```

図3.3 .dsp ファイルの変換ファイル(*.h)

④ホストプログラムの作成

DSPマクロを使用するために、プログラム上で以下の設定が必要になります。

- 1)*.h ファイルのインクルード設定(*.hファイル = I-RAMへの転送元データ)
- 2)DSPマクロのI-RAM/X-RAM/Y-RAMへのデータ転送
- 3)DSPマクロの初期設定/動作開始/演算結果呼び出し等の処理

1) *.h ファイルのインクルード設定

DSPマクロの実行命令コードをインクルードするために、const領域としてセクションを定義します。

図3.5に、アセンブラでの定義例を示します。

```

.section  DSP_CODE_DATA,const,align=2      ;DSPコードデータ
#include "DSP_macro_source.h"

```

図3.5 *.h ファイルのインクルード

2)DSPマクロのI-RAM/X-RAM/Y-RAMへのデータ転送

3)DSPマクロの初期設定/動作開始/演算結果呼び出し等の処理

上記の一連の手順例を下記に示します。

(i)DSP の停止を確認

DSP のDSP-CSRレジスタのRunDSPビットを判定し、DSP が停止していることを確認します。

DSP 動作中は各 RAM 領域へのアクセスができません。

(ii)DSP のコードデータ(*.h)を I-RAM に転送

ASM コンバータが生成したコードデータ(*.h)を、I-RAM に転送します。

(iii)DSP-PC(プログラムカウンタ)の設定

実行を開始する DSP プログラムの先頭アドレス(00H)を設定します。

(iv)変数および係数データを DSP の Y-RAM,X-RAM に転送

係数データ(例では K0、K1、K2)を、X-RAM のアドレス 00H から 02H に転送します。

変数データ(例では R,G,B)のデータを、Y-RAM のアドレス 03H から 05H に転送します。

(v)演算開始

DSP-CSR レジスタの GoDSP ビットを”1“に設定し、DSP を起動します。

これら、(i) ～ (v) の処理を行うプログラム例はサンプルプログラムをご参照下さい。

これらの記述はアセンブラ、C 言語いずれでも対応可能です。

(例では混載された記述にしています。C 記述の main()関数がプログラムスタートとなります。)

<参考>サンプル DSPプログラムの内容

*.dspファイルでは、下記の演算を実行するよう記述しています。

(計算式)

$$Y = K0 \cdot R + K1 \cdot G + K2 \cdot B$$

K0、K1、K2:係数データ → X-RAMに格納します。

R、G、B:変数データ → Y-RAMに格納します。

Y:演算結果データ → Y-RAMに格納します。演算結果はSTR命令で指定したフォーマットで格納されます。

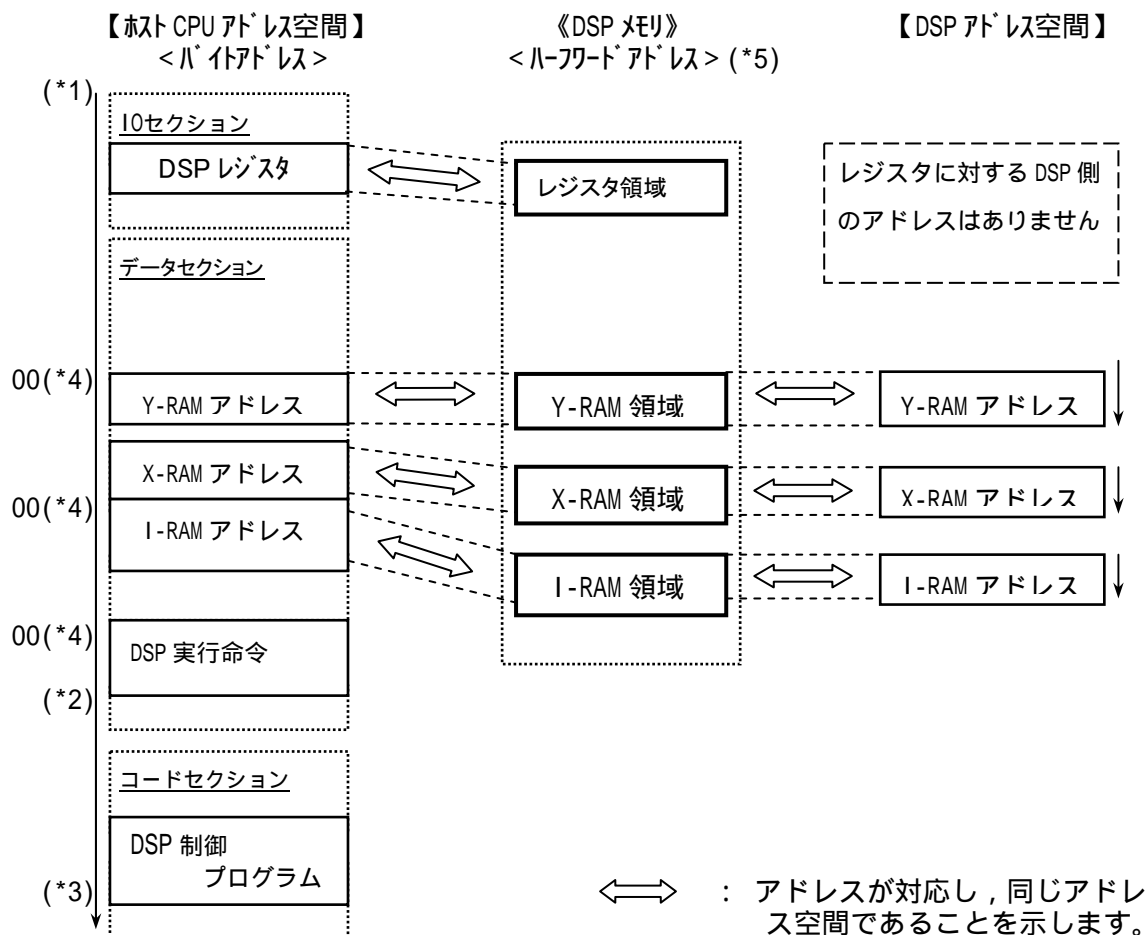
(アセンブラ記述)

```
MUL X_K0,Y3_R
MAC X_K1,Y4_G
MAC X_K2,Y5_B
STYIH Y0_Y,Q8 ;割り込み発生
```


付録

A メモリの配置

DSP 系におけるホスト CPU プログラム, および DSP 自身から見たメモリ配置の概要を以下に示します. アドレスの詳細は, 各製品のマニュアルを参照して下さい。



(*1) DSP のレジスタに相当する領域を,

IO セクションに割り当てます。

(*2) DSP の I-RAM へ転送する DSP の実行命

令であり, ASM コンバータが出力したデータ
です。(*h)

通常, ここから DSP の I-RAM 領域に
転送したのち, 実行します。

(*3) DSP の制御および割り込み処理を行う
手続き部分です。

(*4) DSP 側では, 独自のアドレスを持っており,

X-RAM, Y-RAM, I-RAM それぞれがアドレス 0 から
始まります。

例えば, I-RAM の DSP 実行命令で X-RAM の先
頭ワードを参照したとき, アドレスは 0 になり
ます。同様に, Y-RAM の先頭を参照したときの
アドレスも 0 です。

(*5) 2 バイト (16 ビット) 単位のアドレスです。

(注記) ホスト CPU のプログラムから DSP の X-RAM, Y-RAM, I-RAM 領域を参照する場合, すなわち, ホスト CPU のアドレスによりレジスタ以外の DSP 領域を参照する場合は, 必ず, 偶数アドレスを指定し, かつハーフワード (16 ビット)またはワード (32 ビット)単位でアクセスして下さい。

B MB91260B シリーズ DSPメモリ空間の例

MB91260B シリーズの DSP のメモリ空間の例を以下に示します。

この DSP メモリ空間は, ホスト CPU 側から見たときのアドレスと DSP として動作するときのアドレスを持っています。

ホストCPU側から

DSPとして動作すると

見たアドレス

きのアドレス

Address0003A0H	DSP-PC(プログラムカウンタ)	DSP-CSR(コントロール/ステータス)	Read/Write
0003A2H	DSP-LY(遅延レジスタ)上位	DSP-LY(遅延レジスタ)下位	Read/Write
0003A4H	DSP-OT0(出力キュー0)上位	DSP-OT0(出力キュー0)下位	Read
0003A6H	DSP-OT1(出力キュー1)上位	DSP-OT1(出力キュー1)下位	Read
0003A8H	DSP-OT2(出力キュー2)上位	DSP-OT2(出力キュー2)下位	Read
0003AAH	DSP-OT3(出力キュー3)上位	DSP-OT3(出力キュー3)下位	Read
0003ACH	予約領域	予約領域	アクセス禁止
0003AEH	予約領域	予約領域	アクセス禁止
0003B0H	DSP-OT0(出力キュー0)上位	DSP-OT0(出力キュー0)下位	Read
0003B2H	DSP-OT1(出力キュー1)上位	DSP-OT1(出力キュー1)下位	Read
0003B4H	DSP-OT2(出力キュー2)上位	DSP-OT2(出力キュー2)下位	Read
0003B6H	DSP-OT3(出力キュー3)上位	DSP-OT3(出力キュー3)下位	Read
Address00C000H	X-RAM (係数 RAM) 64 × 16bit		00H (Read/Write) 3FH
00C07EH			
Address00C080H	Y-RAM (係数 RAM) 64 × 16bit		00H (Read/Write) 3FH
00C0FEH			
Address00C080H	I-RAM (係数 RAM) 256 × 16bit		00H (Read/Write) FFH
00C100H			

(注記) X-RAM/Y-RAM/I-RAM について, ホスト CPU 側からみたアドレス, DSP として動作するときのアドレスおよび容量については, 製品のマニュアルを参照して下さい。

(以上)